

# Stochastic optimization for optimal and model-predictive control

J. R. Banga<sup>1</sup>, R. Irizarry-Riverat<sup>2</sup> and W. D. Seider<sup>2\*</sup>

<sup>1</sup> Chemical Engineering Laboratory, IIM-CSIC, Eduardo Cabello 6, 36208 Vigo, Spain

<sup>2</sup> Department of Chemical Engineering, University of Pennsylvania, Philadelphia, PA 19104-6393, USA

† Currently with DuPont Electronics, Manati, Puerto Rico 00674.

\* Author to whom correspondence should be addressed

## Abstract

The integrated-controlled-random-search for dynamic systems (ICRS/DS) method is improved to include a moving-grid strategy and is applied to more challenging problems including: (1) the optimal control of a fed-batch bioreactor, a plug-flow reactor exhibiting a singular arc, the van der Pol oscillator; and (2) the model-predictive control (MPC) of the Czochralski (CZ) crystallization process. This technique has several advantages over the gradient-based optimization methods with respect to convergence to the global optimum and the handling of singular arcs and non-differentiable expressions. Furthermore, its implementation is very simple and avoids tedious transformations that may be required by other methods.

In MPC, a nonlinear program is solved to adjust the manipulated variables so as to minimize a control objective. The major difficulty in MPC implementation is in the handling of the dynamic constraints. The ICRS/DS method is applied for the control of the CZ crystallization process and is shown to be an attractive alternative to: (1) sequential integration and optimization, (2) the use of finite element/orthogonal collocation to convert the ODEs to algebraic constraints, and (3) successive linearization of the ODEs.

## 1. Introduction

The nonlinear characteristics of most chemical processes, especially batch processes, present challenging optimal control problems. Examples arise in the scheduling of batch

processes, the dynamic optimization of batch reactors, and the determination and tracking of process start-up and shut-down trajectories (Terwiesch et al., 1994; Chang and Hsieh, 1995).

In the model-predictive control (MPC) strategy, disturbances are rejected by adjusting the manipulated variables several sampling intervals into the future to minimize a control objective. The formulation of the optimization problem is similar to that for optimal control, with several important differences. First, in MPC, the input function,  $u\{t\}$ , is restricted to discrete step functions. Furthermore, the optimization problem is solved repeatedly to determine the control moves for successive sampling intervals. Also, in MPC, to reduce the process/model mismatch, parameter estimation is implemented through the solution of a companion nonlinear program, as summarized in Fig. 1. At the beginning of the first sampling interval ( $t = 0$ ), the vector of parameters is initialized at  $p_{c,1}$ . To select the vector of manipulated variables,  $u_{l,k}$ , the first nonlinear program (Controller 1) is solved. Note that  $k$  is the sample counter at  $t = 0$ ; also, the nonlinear program is presented in the section on Model-Predictive Control. Then, at the end of the first sampling interval of duration  $T$ , the measurements of the state variables,  $x_{l,k+1}^p$  are compared with the model predictions,  $x_{l,k+1}$ , and new parameters,  $p_{c,2}$ , are estimated. The sample counter is incremented ( $k \leftarrow k + 1$ ) and this procedure is repeated at the subsequent sampling instants. As more data are collected, the parameter estimator becomes more effective in reducing the process/model mismatch.

## 2. Optimal control

The optimal-control problem (OCP), for a fixed terminal time, can be stated as:

Find  $u\{t\}$  over  $t \in [t_0, t_f]$  to minimize (or maximize):

$$J[x,u] = \theta[x, \{t_f\}] + \int_0^t \phi[x\{t\}, u\{t\}, t] dt \quad (1a)$$

subject to:

$$\frac{dx}{dt} = \Psi[x\{t\}, u\{t\}, t] \quad (1b)$$

$$x\{t_0\} = x_0 \quad (1c)$$

$$h[x\{t\}, u\{t\}] = 0 \quad (1d)$$

$$g[x\{t\}, u\{t\}] \leq 0 \quad (1e)$$

$$x^L \leq x\{t\} \leq x^u \quad (1f)$$

$$u^L \leq u\{t\} \leq u^u \quad (1g)$$

where  $J$  is the performance index,  $x$  is the vector of state variables,  $u$  the vector of control variables, (1b) is the system of ordinary-differential equality constraints, (1c) are the initial conditions for (1b), (1d) and (1e) are the equality and inequality algebraic constraints, and (1f) and (1g) are the upper and lower bounds for the state and control variables.

Although many algorithms have been proposed for the predictive control of constrained linear systems, the development of algorithms for the solution of the general OCP, with nonlinear constraints involving the state variables, has received much less attention. Basically, there have been two deterministic approaches (Tanartkit and Biegler, 1995): the calculus of variations (Pontryagin's maximum principle) and transformation methods, in which the original OCP is transformed into a nonlinear program (NLP). However, the calculus of variations is not effective for state-constrained systems.

Transformation methods can be classified as sequential or simultaneous. In the sequential strategy, the manipulated variables are parameterized (i.e. represented by polynomials in time with unknown coefficients) and the differential-algebraic equations (DAEs) are integrated using the coefficients adjusted in each iteration of an optimization algorithm (Vassiliadis, 1993). In the simultaneous strategy (Biegler, 1984; Logsdon and Biegler, 1989; Tanartkit and Biegler, 1995; Tieu et al., 1995), both the unknowns of the DAEs (state and other variables) and the coefficients of the polynomials for the manipulated variables are adjusted simultaneously to satisfy the stationarity conditions.

Luus (1990, 1992, 1994, 1995) computed improved performance indices, compared with those obtained with these strategies, through the use of penalty functions and his iterative dynamic programming (IDP) method. He also demonstrated global convergence properties that are not shared by these gradient-based approaches. Note, however, that the convergence properties of the IDP and gradient-based methods are closely related to the number and position of the discretization intervals. Dadebo and McAuley (1995a, 1995b), recently refined this method using penalty functions formulated using absolute errors.

The ICRS/DS (Integrated Controlled Random Search for Dynamic Systems) algorithm has been used by Banga et al. (1991, 1994) and Banga and Seider (1996) to solve efficiently several complex OCPs. The main advantage of this method is that convergence is assured, independent of the starting control profiles, adding reliability in the solution of multimodal problems; that is, while convergence to the global optimum cannot be guaranteed, the case studies by Banga and Seider (1996) and herein demonstrate excellent convergence properties. Furthermore, it is simple to implement, as no transformation of the original OCP is required. Finally, it has been shown to be computationally faster than IDP for the solution of several case studies Banga and Seider (1996).

In this paper, the ICRS/DS algorithm is extended to include a moving-grid strategy. Also, the algorithm is stated more completely and explicitly. Solutions are presented, for the first time, of three optimal-control problems involving singular arcs. In addition, the solution of a complex, nonlinear MPC to reject the disturbances in the Czochralski (CZ) crystallization process is presented.

### 3. Model-predictive control (MPC)

Nonlinear MPC is attractive for batch processes, which operate over a wide range of conditions, because the nonlinear model of the process is optimized without linearization. The nonlinear model is utilized to determine the future control moves, as well as to update the model parameters. As discussed above, these features are illustrated in Fig. 1.

At each sampling instant  $k$ , the nonlinear model predicts the output vector,  $x$ , at each of  $P$  sampling instants in the future as a function of the vector of manipulated inputs,  $u$ ,  $M$  sampling instants in the future, as shown in Fig. 2. The manipulated inputs are adjusted to minimize the control objective in solution of the following NLP:

Find  $u_k, \dots, u_{k+M-1}$  to minimize:

$$J\{x, u\} = \sum_{j=1}^P \left[ y_j^2 \left( x\{t_{k+j}\} - x^{SP}\{t_{k+j}\} \right)^T \right]$$

$$C(x\{t_{k+j}\} - x^{SP}\{t_{k+j}\}) + \beta_j^2(u_{k+j-1} - u_{k+j-2})^T D(u_{k+j-1} - u_{k+j-2}) \quad (2a)$$

subject to:

$$\frac{dx}{dt} = \Psi[x\{t\}, u\{t\}, p\{t\}, t] \quad (2b)$$

$$x(t_k) = x_k \quad (2c)$$

$$h[x\{t\}, u\{t\}, p\{t\}] = 0 \quad (2d)$$

$$g[x\{t\}, u\{t\}, p\{t\}] = 0 \quad (2e)$$

$$x^L \leq x\{t\} \leq x^U \quad (2f)$$

$$u^L \leq u_k \leq u^U \quad \forall k \quad (2g)$$

At sampling instant,  $k$ , the manipulated variables  $M$  sampling instants in the future,  $u_k, \dots, u_{k+M-1}$ , are adjusted to minimize the weighted sum of the squares of the predicted deviations of the state variables from their set points and the changes in the manipulated variables. Given estimates of the parameters of the model,  $p\{t\}$ , the state variables are predicted  $P$  sampling instants in the future by integration of the DAEs ((2b)-(2d)). Note that  $c$  is a vector of  $n$  weights on the state variables,  $C = \text{diag}\{c\}$ ,  $d$  is a vector of  $m$  weights on the manipulated inputs,  $D = \text{diag}\{d\}$  and  $y_j$  and  $\beta_j$  are weighting coefficients for each sampling instant over the predictive horizon,  $P$ . During each sampling interval, only  $u_k$  is implemented. However, the values of  $u_k, \dots, u_{k+M-1}$ , and  $x\{t_{k+1}\}, \dots, x\{t_{k+p}\}$  are useful in projecting the performance of the controlled system and providing early warnings when the system is approaching process design and operating constraints. Typical variations of the input and output variables, in response to a disturbance, are illustrated over the time horizons in Fig. 2. Note that in batch processes, the operating trajectory,  $x^{SP}\{t\}$ , usually varies in time.

The major difficulty in MPC is in the solution of the dynamic constraints. Thus far, three gradient-based approaches have been implemented: (1) sequential integration and optimization, (2) the use of finite element/orthogonal collocation to convert the ODEs to algebraic constraints, and (3) successive linearization of the ODEs. In this manuscript, the ICRS/DS method is shown to be an attractive alternative.

#### 4. ICRS/DS: a stochastic algorithm for dynamic optimization

In the ICRS/DS algorithm, the OCP is transformed into a constrained NLP using a simple, yet flexible, parameterization of the control function, as described below. The constrained NLP is solved by adjusting the coefficients of the polynomial approximation to the control function using the stochastic ICRS method, with the system of DAEs solved at each iteration using an efficient and reliable initial-value integrator (e.g. DASSL). With this approach, no modifications of the DAEs are required. The use of the ICRS algorithm enforces convergence to the global optimum, easily and effectively handling discrete changes and non-differentiable functions.

The control function,  $u\{t\}$  over  $t \in [t_0, t_f]$  is parameterized using variable-length, piecewise-linear polynomials, defined using  $N$  grid points  $(\theta_i, \omega_i, i = 1, \dots, N)$ . For iteration  $k$ , within the optimization procedure, the polynomials are:

$$u^k\{t\} = \frac{\omega_{i+l}^k - \omega_i^k}{\theta_{i+l}^k - \theta_i^k} \cdot (t - \theta_i^k) + \omega_i^k \quad (3)$$

Where  $\theta_i^k \leq t \leq \theta_{i+l}^k$ . The resulting NLP involves  $2N \cdot M$  unknown coefficients  $(\theta_i, \omega_i)$ , where  $M$  is the number of control variables. Therefore, the vector of decision variables,  $\xi$ , for each control variable in iteration  $k$  of the ICRS/DS algorithm is:

$$\begin{aligned} \xi_i^k &= \theta_i^k, i = 1, \dots, N \\ \xi_i^k &= \omega_{i-N}^k, i = N + 1, \dots, 2N \end{aligned} \quad (4)$$

Since the ICRS/DS algorithm requires the explicit declaration of upper and lower bounds for the decision variables, constraints for the control vector are:

$$\begin{aligned} \xi_i^U &= \theta_i^U, \xi_i^L = \theta_i^L, i = 1, \dots, N \\ \xi_i^U &= \theta_{i-N}^U, \xi_i^L = \theta_{i-N}^L, i = N + 1, \dots, 2N \end{aligned} \quad (5)$$

In the original version of the ICRS/DS algorithm, a fixed and constant discretization ratio was used taking:

$$\theta_i^U = \theta_i^L + \varepsilon = \left( \frac{t_f}{N-1} \right) \cdot (i-1) + \varepsilon; i = 1, \dots, N \quad (6)$$

where  $\varepsilon$  is a small number, with a large number of grid points necessary to approximate non-smooth control profiles. Herein, the efficiency of the discretization is improved

through the use of moving-grid points. As described below, the bounds, and,  $\theta_i^L$  and  $\theta_i^U$   $i = 1, \dots, N$  are adjusted at each iteration. In this way, complex control profiles can be approximated with sufficient accuracy using fewer grid points, which increases both the rate of convergence and the reliability of the method.

In MPC, the control function is comprised of discrete steps over the manipulated variable horizon. At each sampling instant,  $k$ , NLP (2a)-(2g) is solved to determine:

$$u\{t_k\} = \begin{cases} u_k \\ u_{k+1} \\ \vdots \\ u_{k+M-1} \end{cases} \quad (7)$$

#### 4.1. The ICRS/DS algorithm

This algorithm is a modification of the ICRS method (Banga and Casares, 1987; Casares and Banga, 1989), a generalization of the method originally proposed by Gouleber and Casares (1978) for the optimization of processes in the steady state. The algorithm involves a parameter that adjusts the search window and the ability to locate the global minimum of multi-modal objective functions, as described by Banga and Seider (1996). Two additional parameters ( $k_2$  and  $n_e$ ) control the rate of convergence to a stationary point. These parameters are defined in the improved ICRS/DS algorithm that follows, which introduces flexibility in the parameterization of the manipulated variable:

Step 1 -- Read parameters and input data, including the upper and lower bounds for the decision variables,  $\xi$ , and the initial vector of decision variables,  $\xi^0$ , which must be feasible. When no initial vector is available, ICRS/DS generates it using random searches from a uniform probability distribution over each of the decision variables, stopping after a feasible vector is found.

Step 2 -- Evaluate the performance index at the initial vector,  $J_0 = J\{\xi^0\}$ , and check the inequality constraints, as follows:

Step 2a -- Using the parameterization of the control variables above, solve the ODEs, beginning with their initial values, together with the so-called algebraic constraints, using DASSL.

Step 2b -- Check the inequality constraints, and the bounds on the state variables. When any one is not satisfied, the decision vector is not feasible. Stochastic optimization for optimal and model-predictive control

Step 2c -- When the decision vector is feasible, calculate the performance index.

Step 3 -- when the initial vector is infeasible, return to Step 1 and either stop or generate a new vector.

Step 4 -- For each decision variable, select the smallest subinterval:

$$\lambda_i^k = \min \left\{ \left( \xi_i^U - \xi_i^k \right), \left( \xi_i^k - \xi_i^L \right) \right\}, i = 1, \dots, 2N \quad (8)$$

Step 5 -- Calculate the vector of standard deviations, using the heuristic parameter  $k_1 = 1/\beta$ :

$$\sigma_i^k = k_1 \cdot \lambda_i^k, i = 1, \dots, 2N \quad (9)$$

Step 6 -- Generate a new decision vector using a Gaussian distribution with standard deviation,  $\sigma_i^k$ , and mean at  $\xi_i^k$ :

$$\xi_i^{k+1} = \xi_i^k + \sigma_i^k \cdot \Gamma_i, i = 1, \dots, 2N \quad (10)$$

where  $\Gamma$  is a vector of pseudo-random numbers generated using a Gaussian distribution having a mean at zero and standard deviation at unity. When  $\xi_i^{k+1} \leq \xi_i^L$  or

$\xi_i^{k+1} > \xi_i^U$ ,  $\xi_i^{k+1}$  is rejected and another value is generated.

Step 7 -- Evaluate the performance index,  $J^{k+1}$  at the new decision vector, following Steps 2a-c. When this vector is feasible and the performance index has improved ( $J^{k+1} < J^k$  for minimization), skip to Step 9. Otherwise, carry out Step 8.



Step 8 -- Increment the failure counter,  $F$ . When,  $F > n_e \cdot 2N$ ,  $\sigma_i^k \leftarrow k_2 \cdot \sigma_i^k$ ,  $i = 1, \dots, 2N$  and reset the counter  $F$ , where  $n_e = 4$  and  $k_2 = 1/2$ . Return to Step 6.

Step 9 -- Check the convergence criteria:

$$\frac{|\xi_i^{k+1} - \xi_i^k|}{\xi_i^U - \xi_i^L} < tol_\xi \quad (11)$$

and

$$\frac{|J^{k+1} - J^k|}{J^k} < tol_J \quad (12)$$

and stop when satisfied. Otherwise, recalculate the bounds on the discretization intervals for the independent variable:

$$\xi_{,}^{L,k} = \xi_i^k - \left( \frac{\xi_i^k - \xi_{i-1}^k}{2} \right) \cdot \beta; i = 1, \dots, N \quad (13)$$

$$\xi_{,}^{U,k} = \xi_i^k + \left( \frac{\xi_{i+1}^k - \xi_i^k}{2} \right) \cdot \beta; i = 1, \dots, N \quad (14)$$

where  $\beta \in [0, 1]$ , with good results obtained using  $\beta = 0.95$ . Then, increment the iteration counter such that  $\xi_i^k = \xi_i^{k+1}$  and  $j^k = j^{k+1}$ , reset counter  $F$ , and return to Step 4.

#### 4.2. Implementation of the ICRS/DS algorithm

Solution of an optimal control problem by the ICRS/DS algorithm is straightforward. Only functions or subroutines are needed to evaluate the DAEs, check the inequality constraints, and calculate the performance index.

There is no need to transform the original problem, even when it is poorly scaled. Penalty functions are not needed because the constraints are handled directly. Since the method is not gradient-based, non-differentiable systems can be solved without transformations. Hence, the ICRS/DS algorithm is implemented quickly with little effort.

It should be noted that the values of  $k_1$ ,  $n_e$ , and  $k_2$  reported in the algorithm above have been used successfully to solve many optimal control problems. These parameters are tuned to provide better performance when solving problems with special characteristics.

For example, Banga and Seider (1996) set  $k_1 = 2$  to obtain the global minimum of a multimodal objective function in the calculation of liquid-liquid equilibria.

When solving high-index systems, care must be taken in the use of DASSL. To avoid spurious results, Feehery et al. (1995) derive an Index-1 system of DAEs.

## 5. Optimal control case studies

### 5.1. van der Pol oscillator

This oscillator has been studied by Vassiliadis (1993) and Tanartkit and Biegler (1995).

The optimization problem is:

$$\text{Min } J = y_3\{t_f\}$$

S.T.:

$$\frac{dy_1}{dt} = (1 - y_2^2)y_1 - y_2 + u \quad (\text{I.1})$$

$$\frac{dy_2}{dt} = y_1 \quad (\text{I.2})$$

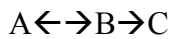
$$\frac{dy_3}{dt} = y_1^2 + y_2^2 + u^2 \quad (\text{I.3})$$

$$-0.3 \leq u\{t\} \leq 1.0 \quad (\text{I.4})$$

with the initial condition  $y\{0\} = [0.0 \ 1.0 \ 0.0]^T$ . The authors above solved the OCP for  $t_f = 5$ .

### 5.2. Case Study II. Plug-flow reactor with a singular arc

A plug-flow reactor, involving the reactions,



is packed with a mixture of two catalysts, each specific to one of the reactions. It is desired to adjust the fraction of the first catalyst (specific to the reversible reaction) so as to maximize the production of species C. This is accomplished by solving the OCP, initially formulated by Jackson (1968):

$$\text{Max}_{u\{z\}} J = 1 - x_A\{z_f\} - x_B\{z_f\}$$

S.T.:

$$\frac{dx_A\{z\}}{dz} = u\{z\} \cdot [10x_B\{z\} - x_A\{z\}] \quad (\text{II.1})$$

$$\frac{dx_B\{z\}}{dz} = -u\{z\} \cdot [10x_B\{z\} - x_A\{z\}] - [1 - u\{z\}] \cdot x_B\{z\} \quad (\text{II.2})$$

$$0 \leq \{z\} \leq 1 \quad (\text{II.3})$$

with the initial conditions  $x_A\{0\} = 1.0$ ,  $x_B\{0\} = 0$ , where  $x_A$  and  $x_B$  are the mole fractions of species A and B, and  $u\{z\}$  is the fraction of catalyst associated with the reversible reaction at position  $z$  in the reactor of length unity. To circumvent the singular arc, reported by Logsdon and Biegler (1989) and Dadebo and McAuley (1995b), the former authors apply a high-order collocation method.

### 5.3. Case Study III. Optimal production of secreted protein in a fed-batch reactor

This OCP, which has been studied by Park and Ramirez (1988), Luus (1992), Yang and Wu (1994), and Luus (1995), maximizes the secreted heterologous protein in a fed-batch bioreactor. Its dynamic model accounts for host-cell growth, gene expression, and the secretion of expressed polypeptides (Park and Ramirez, 1988).

$$\underset{u}{Max} \ J = x_1\{t_f\} \cdot x_5\{t_f\}$$

S.T.:

$$\frac{dx_1}{dt} = g_1 \cdot (x_2 - x_1) - \frac{u}{x_5} x_1 \quad (\text{III.1})$$

$$\frac{dx_2}{dt} = g_2 x_3 - \frac{u}{x_5} x_2 \quad (\text{III.2})$$

$$\frac{dx_3}{dt} = g_3 x_3 - \frac{u}{x_5} x_3 \quad (\text{III.3})$$

$$\frac{dx_4}{dt} = -73 \cdot g_3 x_3 + \frac{u}{x_5} \cdot (20 - x_4) \quad (\text{III.4})$$

$$\frac{dx_5}{dt} = u \quad (\text{III.5})$$

$$g_1 = \frac{4.75g_3}{0.12 + g_3} \quad (\text{III.6})$$

$$g_2 = \frac{x_4 \cdot e^{-5x_4}}{0.1 + x_4} \quad (\text{III.7})$$

$$g_3 = \frac{21.87 \cdot x_4}{(x_4 + 0.4)(x_4 + 62.5)} \quad (\text{III.8})$$

$$0 \leq u \leq 2 \quad (\text{III.9})$$

with the initial condition  $x\{0\} = [0 \ 0 \ 1 \ 5 \ 1]^T$  and the final time,  $t_f=15$  h. Here,  $x_1$  and  $x_2$  are the concentrations ( $L^{-1}$ ) of the secreted protein and the total protein, respectively,  $x_3$  is the glucose concentration ( $g \ L^{-1}$ ),  $x_4$  is the substrate concentration ( $g \ L^{-1}$ ),  $x_5$  is the holdup volume (L), and  $u$  is the feed flow rate ( $L \ h^{-1}$ ).

## 6. MPC case study: Czochraiski (CZ) crystallization process

In the CZ crystallization process, single crystals are grown from the melt by pulling a seed crystal very slowly while maintaining a desired thermal environment in the furnace, as illustrated in Fig. 3a. The objectives are to achieve a constant radius, with low concentration of impurities and dislocations, and a uniform dopant distribution. CZ crystallizers are nonlinear batch processes whose variables have widely differing length and time scales.

Irizarry-Rivera and Seider (1997a, 1997b) propose a control strategy for this system, consisting of two coupled MPCs, operating at different time and length scales, as illustrated in Fig. 3b. The first, referred to as a capillary controller, controls the radius of the crystal by manipulating the pulling velocity. The second, referred to as a bulk controller, manipulates the power inputs to the heaters (and heat pipes) to control the pulling velocity, and the microscopic control objectives. The details of the design, including the control objectives, the manipulated inputs and process outputs, a model to relate the process outputs to the manipulated inputs, and inequality constraints to satisfy the microscopic control objectives, are not repeated here.

Those manuscripts present dynamic models for the capillary and bulk controllers. The model for the capillary controller relates the radius to the pulling velocity using the wetting conditions and a heat balance at the melt crystal/interface. For the bulk controller, two models are presented that relate the heat flux at the melt/ crystal interface to the power inputs. When conduction dominates the heat transfer in the melt, the boundary-element technique is utilized to reduce the dimensionality of the system. When convection dominates, a reduced-order model is utilized.

## 7. Results and discussion

### 7.1. Optimal control problems

All of the results were computed using  $N = 6 - 10$  grid points for the independent variable, and are in agreement with the results computed using the original ICRS/DS algorithm (Banga et al., 1991, Banga et al., 1994).

Case study I. For the van der Pol oscillator, no convergence problems were encountered. Starting with a constant control profile,  $u = 0.5$ , discretized using four finite elements, a minimum of  $J = 2.8715$  was computed, with CPU times ranging between 5 and 10 min, using a PC i486/50 MHz. The optimal control profile is shown in Fig. 4. These results compare well with those reported by Vassiliadis (1993;  $J = 2.8681$ , using 10 finite elements) and Tanartkit and Biegler (1995;  $J = 2.8710$ ). The former reports CPU times between 44 and 450 s using a Sun SPARC 2 workstation, while the latter report CPU times of 36 s using a DECstation 5000. Applying the LINPACK benchmarks for FORTRAN 77 (PC i486/50 MHz = 1.3 Mflops, SPARC 2 = 4.0 Mflops, and DEC-station 5000 = 5.3 Mflops) the transformed CPU times for the PC are:

ICRS/DS algorithm	5-10 min
Sequential algorithm	2.2-22.5 min
Simultaneous algorithm	2.4 min.

Clearly, these CPU times are comparable. Furthermore, when the ICRS/DS algorithm is implemented with 10 finite elements,  $J = 2.87$ , and the CPU times are between 11 and 15 min.

Case study II. For the plug-flow reactor, starting with  $u = 0.5$ , the ICRS/DS algorithm computed a maximum of  $J = 0.04805$ , with the optimal control profile in Fig. 5. These results agree with those reported by Dadebo and McAuley (1995b) using the IDP method. However, CPU times of 10 min. (i486 $\beta$ 3) are required, while the CPU times

using the ICRS/DS algorithm varied between 40 and 100 s. This advantage was observed also by Banga and Seider (1996).

Case study III. For the fed-batch reactor, Luus (1995) reported a very low sensitivity of the control policy on the yield, which often leads to computational difficulties, especially using gradient-based algorithms for optimal control. Using the ICRS/DS algorithm, starting with  $u = 0.5$ , an optimum of  $J = 32.562$ , with the optimal control profile in Fig. 6, was computed easily with CPU times between 12 and 35 min. Using IDP, Luus (1992) reports  $J = 32.64$ , and Luus (1995) reports  $J = 32.686$ , values slightly greater than that computed herein. However, CPU times of 2.3 h (i486/66) are reported for each pass of the IDP, with the maximum obtained in 15 passes.

To evaluate the advantages of the moving-grid discretization, case study I was solved using a fixed grid, as in the original ICRS/DS algorithm. In all calculations, lower values of the performance index were determined ( $J = 2.89 - 2.93$  using a fixed grid). Similar results were obtained for case studies II and III. To illustrate the behavior of the moving-grid for case study I, with  $\beta = 0.6$ , the bounds,  $\xi_i^{U,k}$  and  $\xi_i^{L,k}$ , and the corresponding grid points  $\xi_i^k$  are plotted as a function of the iteration number,  $k$ , in Fig. 7. Note that to achieve comparable values of the performance index when using a fixed grid, the number of grid points must be increased, with a substantial increase in the CPU times. When the manipulated variable varies smoothly, the fixed and moving-grid strategies perform similarly. The moving-grid strategy is most effective when the manipulated variable experiences abrupt changes and sharp fronts.

Note that, for case study I,  $\beta$  in the range of 0.4-0.95 provided almost equivalent results, although somewhat shorter CPU times were recorded for  $\beta = 0.6$ . Similar results were obtained for case studies II and III. This insensitivity of the algorithm's performance is because  $\beta$  controls only the upper and lower bounds on  $\xi$ , which change at each iteration. Since this behavior could be problem dependent, additional case studies are needed.

## 7.2. MPC of CZ crystallization process

Results are presented for two cases involving the crystallization of silicon in a laboratory puller with the operating parameters in Table 1. In the first, the capillary controller adjusts the pulling velocity in response to a disturbance in the crystal radius, and in the second, the bulk controller, assuming a conduction-dominated melt, adjusts the average pulling velocity by manipulating the temperature of the heater (not the power input as shown in Fig. 3 and implemented by Irizarry-Rivera and Seider (1997a, 1997b)). Note that, for both cases, the predicted and manipulated-variable horizons are identical; that is,  $M = P$ .

Figure 8 shows the response to a disturbance that increments the crystal radius by 0.4 mm at time zero. As seen in Fig. 8a, most of the error is eliminated in about one minute. The model is a lumped-parameter energy balance (involving the dynamic contact angle at the tri-junction between the crystal, melt, and vapor surroundings), and hence, the optimization is completed in Results are presented for two cases involving the crystallization of silicon in a laboratory puller with the operating parameters in Table 1. In the first, the capillary controller adjusts the pulling velocity in response to a approximately 1-2 s on a DECAlpha 3000 workstation. Note that  $M = P = 3$  and the sampling interval,  $T$ , is 15 s.

For the bulk controller, the heater temperature is the manipulated variable. With the melt dominated by conduction, its boundary is discretized using 68 constant elements: 18 along the crucible bottom, 18 along the side wall, 8 along the free surface, 8 along the free surface approaching the tri-junction, and 8 along the melt-crystal interface. Note that the refined grid in the vicinity of the tri-junction is required to compute sufficiently accurate temperatures; alternatively, higher-order trial functions could be incorporated. Furthermore, the transient terms are discretized using the Dual Reciprocity Theorem with five internal poles added to provide increased accuracy. Using the ICRS/DS method, the number of decision variables,  $M$ , is considerably less than the number required to implement a finite-element discretization with a successive-quadratic programming algorithm (i.e.  $M$  plus the number of boundary and internal nodes), as implemented, for example, by Cuthrell and Biegler (1987) using orthogonal collocation (in the so-called SQP/OC algorithm). For the latter, projection methods could reduce the number of decision variables to  $M$ , but a large Jacobian matrix (of order 2000, 3000,

and 4000 for  $M = 1, 2$ , and  $3$ , respectively) must be inverted. This is avoided by using the boundary-element method within the ICRS/DS algorithm.

Figure 9 shows the response of the laboratory-scale puller (with nominal operating parameters in Table 1) to a disturbance in which the average pulling velocity is increased by  $2 \text{ cm hr}^{-1}$ . As seen in Fig. 9a, most of the error is eliminated in about 1.5 min with most of the error eliminated in the first two sampling intervals. For industrial-scale pullers, a similar response would be obtained in about 30 min.

The ICRS/DS algorithm was utilized with  $k_1 = 1/2$ ,  $n_e = 4$ , and  $k_2 = 1/2$ , as recommended by Banga et al. (1991). Note that the moving-grid algorithm does not apply because, in MPC, the manipulated variable experiences a step change after each constant sampling interval. For each MPC optimization, with  $M = 1$  and  $T = 30\text{sec}$ , approximately 5-10 CPU seconds were required on a DECalpha 3000 workstation. When  $M$  was increased to 2 or 3, approximately 25-40 and 80-120 CPU seconds, respectively, were required, with similar responses computed. Consequently, the former tuning was used in this work. Larger CPU times would be required for industrial-scale pullers, but their responses are considerably slower as noted above. As workstation speeds increase, the MPC optimizations will be completed earlier in the sampling intervals.

### 7.3. Performance summary

This section summarizes the results of the case studies presented herein, it being noted that quantitative conclusions concerning the performance of methods for the solution of optimal control and nonlinear MPC problems are almost always based upon the results of case studies. To our knowledge, theoretically-based conclusions have not been derived for nonlinear systems which are often endowed with non-differentiable functions. In lieu of theoretically-based conclusions, several researchers (especially Luus et al.) have presented many case studies to evaluate the performance of competitive algorithms.

Results using the ICRS/DS algorithm are an improvement on those obtained with gradient-based algorithms for the case studies presented herein (and many others



examined in our work), because this stochastic algorithm is not misled by gradient information that often leads to rapid convergence to local optima. The ICRS/DS algorithm normally explores a broader portion of the search space, gradually focusing on the regions most likely to contain the global solution. Of course, convergence to a global optimum is not guaranteed, but when the ICRS/DS parameters (especially  $k_1$ ) are set properly, the convergence properties are excellent, especially when compared with gradient-based methods. This was established by Banga and Seider (1996) and is further confirmed by the more challenging problems solved herein.

Finally, to accompany these excellent convergence properties, the ICRS/DS method achieves computation times comparable to those for the gradient-based methods - a remarkable advantage for a stochastic method that is so much easier to implement, especially for problems involving singular arcs and discontinuous functions.

## 8. Conclusions

It is concluded that:

1. The ICRS/DS method is reliable and efficient, when compared with gradient-based algorithms, in solving challenging optimal-control problems. Its convergence properties are excellent, the computation times are competitive, and it is much easier to implement, especially when singular arcs and discontinuities exist.
2. The moving-grid algorithm is especially effective for the solution of optimal-control problems in which the manipulated variables experience discrete changes in their magnitudes and first derivatives.
3. The ICRS/DS algorithm using a boundary-element method is efficient and reliable for the MPC of the CZ crystallization process when conduction dominates the heat transfer in two dimensions. It circumvents the inefficient solution of a continuous dynamic model for heat transfer in response to discrete, step changes in the manipulated variables.

## Acknowledgements

Partial funding provided by the Computer Integrated Engineering Program of the NSF under Grant Nos. DDM-91-14080 and DDM-94-00775 is gratefully acknowledged.

## References

- Banga, J. R. and W. D. Seider, Global optimization of chemical processes using stochastic algorithms. In *Nonconvex Optimization and its Applications -- State of the Art of Global Optimization—Computational Methods and Applications* (Edited by C. A. Floudas and P. M. Pardalos). Kluwer Academic Publ. (1996).
- Banga, J. R. and Casares, J. J. (1987) Integrated controlled random search: application to a wastewater treatment plant model. In *Process Optimization*, IChemE Syrup. Set. 100, 183-192.
- Banga, J. R., Alonso, A. A., Perez-Martin, R. I. and Singh, R. P. (1994) Optimal control of heat and mass transfer in food and bioproducts processing. *Comput. Chem. Engng* 18, S699-S705.
- Banga, J. R., Perez-Martin, R. I., Gallardo, J. M. and Casares, J. J. (1991) Optimization of the thermal processing of conduction-heated canned foods: study of several objective functions. *J. Food Engng* 14, 25-51.
- Biegler, L. T. (1984) Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Comput. Chem. Engng* 8, 243-248.
- Casares, J. J. and Banga, J. R. (1989) Analysis and evaluation of a wastewater treatment plant model by stochastic optimization. *Appl. Math. Model.* 13, 420-424.
- Chang, J.-S. and Hsieh, W.-Y. (1995) Optimization and control of semi-batch reactors. *Ind. Engng Chem. Res.* 34, 545-556.
- Cuthrell, J. E. and Biegler, L. T. (1987) On the optimization of differential-algebraic process systems. *AIChE J.* 33(8), 1257-1270.
- Dadebo, S. A. and McAuley, K. B. (1995a) A simultaneous iterative solution technique for time-optimal control using dynamic programming. *Ind. Engng Chem. Res.* 34, 2077-2083.
- Dadebo, S. A. and McAuley, K. B. (1995b) Dynamic optimization of constrained chemical engineering problems using dynamic programming. *Comput. Chem. Engng* 19(5), 513-525.

- Feehery, W. F., J. R. Banga and P. J. Barton (1995) A novel approach to dynamic optimization of ODE and DAE systems as high-index problems. AIChE Annual Meeting, Miami.
- Goulcher, R. and Casares, J. J. (1978) The solution of steady-state chemical engineering optimization problems using a random search technique. *Comput. Chem. Engng* 2, 33-36.
- Irizarry-Rivera, R. and W. D. Seider (1997a) Model-predictive control of the Czochralski crystallization process: Part I - conduction-dominated melt. *J. Crys. Growth.*, 178, 593-611.
- Irizarry-Rivera, R. and W. D. Seider (1997b) Model-predictive control of the Czochralski crystallization process: Part II - reduced-order convection model. *J. Crys. Growth.*, 178, 612-633.
- Jackson, R. (1968) Optimal use of mixed catalysts for two successive chemical reactions. *J. Optim. Theory Applic.* 2, 1
- Logsdon, J. S. and Biegler, L. T. (1989) Accurate solution of differential algebraic equations. *Ind. Engng Chem. Res.* 28, 1628-1639.
- Luus, R. (1990) Optimal control by dynamic programming using systematic reduction in grid size. *Int. J. Control* 51, 995-1013.
- Luus, R. (1992) On the application of iterative dynamic programming to singular optimal control problems. *IEEE Transac. Autom. Control* 37(11), 1802-1806.
- Luus, R. (1994) Optimal control of batch reactors by iterative dynamic programming. *J. Proc. Control* 4(4), 218-226.
- Luus, R., Sensitivity of control policy on yield of a fed-batch reactor, Presented at IASTED Int. Conf. on Modelling and Simulation, Pittsburgh, PA., April 27-29, 1995.
- Park, S. and Ramirez, W. F. (1988) Optimal production of secreted protein in fed-batch reactors. *AIChE J.* 34(9), 1550-1558.
- Tanartkit, P. and Biegler, L. T. (1995) Stable decomposition for dynamic optimization. *Ind. Engng Chem. Res.* 34, 1253-1266.
- Terwiesch, P., Agarwal, M. and Rippin, D. T. (1994) Batch unit optimization with imperfect modelling: a survey. *J. Proc. Control* 4(4), 238-258.
- Tieu, D., Cluett, W. R. and Penlidis, A. (1995) A comparison of collocation methods for solving dynamic optimization problems. *Comput. Chem. Engng* 19(4), 375-381.

- Vassiliadis, V. (1993). Computational solution of dynamic optimization problems with general differential-algebraic constraints. Ph.D. Thesis, Imperial College, University of London, U.K..
- Yang, R. L. and C. P. Wu, Global optimal control by accelerated simulated annealing. Presented at the First Asian Control Conference, Tokyo, Japan (1994).

Fig. 1. Schematic of MPC algorithm

Fig. 2. Output horizon P and input horizon M for model-predictive control, k is the index of the current sampling instant, x and u are vectors of the output and input variables.

Fig. 3. Control of the Czochralski crystallization process. (a) Schematic of the CZ crystallizer. (b) MPC controllers.

Fig. 4. Optimal control profile for the van der Pol oscillator

Fig. 5. Optimal control profile for the plug-flow reactor.

Fig. 6. Optimal control profile for the fed-batch reactor.

Fig. 7. Grid points,  $\sim$ , with bounds for the van der Pol oscillator:  $\square = \xi^L$ ,  $O = \xi^U$ ,  $\bullet = \xi$

Fig. 8. Response of capillary controller to a 0.4 mm step. increase in the crystal radius.

Fig. 9. Response of bulk controller to a 2 cm hr<sup>-1</sup> step increase in the average pulling velocity.

Table 1. Operating parameters

Crucible radius, $R_x$	8 cm
Crystal radius, $R^{\text{RET}}$	4 cm
Effective ambient temperature, $T_{\text{eff}}$	1430 K
Effective bottom temperature, $T_b$	1430 K
Melt height	8 cm
Pulling velocity, $v^{\text{set}}$	6 cm hr <sup>-1</sup>
Biot number, $Bi - hR_c/k$	0.5
Radiation number, $Ra$	0